CLAIMS:

1. A method for scheduling data packets transported from input-nodes to output-nodes said data packets being associated with a set of ***N*** input-nodes each having a plurality of ***M*** queues each for queuing data packets for routing to one or more corresponding ***M*** output-nodes, said method comprising:

(a) providing at least two clusters of source port modules, each source port module tracking all queues associated with a respective input-node and each cluster relating to a respective subset of available input-nodes such that each input node is associated with a respective one of said clusters,

(b) for each queue in each source port module destined to an available output node, generating a weight reflecting an urgency of said queue to transmit its queued cells towards the corresponding output-node,

(c) for each source port module tracking a serviceable queue, generating at least one request relating to the serviceable queue having highest weight,

(d) accumulating the respective requests of each source port module in the corresponding cluster of source port modules,

(e) for each cluster of source port modules, choosing requests for which:

i) no two requests in the cluster relate to the same input-node, and

ii) for each output-node, the chosen requests have highest weight for said output-node,

(f) collecting requests from all clusters of source port modules, and determining the highest weight request in respect of each output node receiving requests from one or more input nodes,

(g) sending a grant to the input-node associated with the highest weight request,

(h) removing the output-node associated with the said highest weight request from the available output node set,

(i) removing the input-node associated with the said highest weight request from the available input node set, unless the input-node needs to send the highest weight request to one or more additional output-nodes, and

(j) repeating **(a)** to **(i)** as required.

2. The method according to Claim 1, wherein successive phases of an iteration are pipelined so as to allow independent components of a scheduler carrying out the method to operate in parallel.

3. The method according to claim 1, further including at least one filtering operation to improve performance of said method.

4. The method according to Claim 3, wherein the at least one filtering operation comprises:

i) receiving updated snapshots of available input-nodes and of available output-nodes,

ב) removing requests outgoing from the source port module if either:

(1) said request is from an input node which is not a member of the updated snapshot of the available input-nodes, or,

(2) said request is from an input node which is not a member of the updated snapshot of the available output-nodes.

5. The method according to 3, wherein the at least one filtering operation handles information that is important for the generation of requests by the source port modules.

6. The method according to claim 2, further including at least one filtering operation to improve performance of said method.

7. The method according to Claim 6, wherein the at least one filtering operation comprises:

i) receiving updated snapshots of available input-nodes and of available output-nodes,

ג) removing requests outgoing from the source port module if either:

(1) said request is from an input node which is not a member of the updated snapshot of the available input-nodes, or,

(2) said request is from an input node which is not a member of the updated snapshot of the available output-nodes.

8. The method according to 6, wherein the at least one filtering operation handles information that is important for the generation of requests by the source port modules.

9. The method according to Claim 3, wherein the at least one filtering operation handles information relating to the source port module.

10. The method according to Claim 3, wherein the at least one filtering operation is used for total screening of the source port module cluster requests or for "last moment" screening of requests.

11. A scheduler for scheduling data packets transported from input-nodes to output-nodes, said data packets being associated with a set of ***N*** input-nodes each having a plurality of ***M*** queues each for queuing data packets for routing to a corresponding one of ***M*** output-nodes, said scheduler comprising:

at least two clusters of source port modules associated with respective subsets of input nodes for determining a highest weight queue for each input node in the respective subset associated with each cluster of source port modules,

a scheduler core module coupled to all of the clusters of source port modules for determining to which output node to route the highest weight queue from each input node,

a grant unit coupled to the scheduler core module for matching the output-node with the input-node having the highest priority request, and

a switching unit responsively coupled to the grant unit for enabling each input-node to transfer data to the respective output-node matching said input node.

12. The scheduler according to Claim 11, wherein each cluster of source port modules includes a maximum cluster determination unit for choosing the highest-weight-request for each requested output node that was raised by more than one source port module in the source port module cluster.

13. The scheduler according to Claim 11, further including at least one filter to improve scheduling efficiency.

14. The scheduler according to Claim 13, wherein the at least one filter is located in each of the source port modules.

15. The scheduler according to Claim 13, wherein the at least one filter is located after the source port modules.

16. The scheduler according to Claim 12, further including at least one filter to improve scheduling efficiency.

17. The scheduler according to Claim 16, wherein the at least one filter is located in each of the source port modules.

18. The scheduler according to Claim 16, wherein the at least one filter is located after the source port modules.

19. The scheduler according to Claim 12, further including at least one filter located after the maximum cluster determination unit to improve scheduling efficiency.

20. The scheduler according to Claim 11, wherein independent components thereof operate in parallel so as to allow pipelined scheduling.